



USPTO

[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)

 Search: ☒ The ACM Digital Library ☐ The Guide

Patch Manager Run time Instruction

THE ACM DIGITAL LIBRARY


[Feedback](#) [Report a problem](#) [Satisfaction survey](#)
Terms used **Patch Manager Run time Instruction**

Found 10,699 of 184,245

Sort results
by

relevance

Display
results

expanded form

[Save results to a Binder](#)[Search Tips](#)☐ Open results in a new window[Try an Advanced Search](#)[Try this search in The ACM Guide](#)

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

Best 200 shown

Relevance scale ☐ ☐ ☐ ☐ ☐**1** [Instruction packing: Toward fast and energy-efficient instruction scheduling](#)

Joseph J. Sharkey, Dmitry V. Ponomarev, Kanad Ghose, Oguz Ergin

June 2006 **ACM Transactions on Architecture and Code Optimization (TACO)**, Volume 3
Issue 2**Publisher:** ACM PressFull text available: pdf(665.64 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Traditional dynamic scheduler designs use one issue queue entry per instruction, regardless of the actual number of operands actively involved in the wakeup process. We propose Instruction Packing---a novel microarchitectural technique that reduces both delay and power consumption of the issue queue by sharing the associative part of an issue queue entry between two instructions, each with, at most, one nonready register source operand at the time of dispatch. Our results show that this techniqu ...

Keywords: Issue queue, instruction packing, low power**2** [Instruction prefetching of systems codes with layout optimized for reduced cache misses](#)

Chun Xia, Josep Torrellas


May 1996 **ACM SIGARCH Computer Architecture News , Proceedings of the 23rd annual international symposium on Computer architecture ISCA '96**, Volume 24
Issue 2**Publisher:** ACM PressFull text available: pdf(1.65 MB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

High-performing on-chip instruction caches are crucial to keep fast processors busy. Unfortunately, while on-chip caches are usually successful at intercepting instruction fetches in loop-intensive engineering codes, they are less able to do so in large systems codes. To improve the performance of the latter codes, the compiler can be used to lay out the code in memory for reduced cache conflicts. Interestingly, such an operation leaves the code in a state that can be exploited by a new type of ...

3 [Retargetable tools for embedded software: Instruction set compiled simulation: a technique for fast and flexible instruction set simulation](#)

Mehrddad Reshadi, Prabhat Mishra, Nikil Dutt


June 2003 **Proceedings of the 40th conference on Design automation****Publisher:** ACM Press

Full text available:  [pdf\(198.91 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Instruction set simulators are critical tools for the exploration and validation of new programmable architectures. Due to increasing complexity of the architectures and time-to-market pressure, performance is the most important feature of an instruction-set simulator. Interpretive simulators are flexible but slow, whereas compiled simulators deliver speed at the cost of flexibility. This paper presents a novel technique for generation of fast instruction set simulators that combines the benefit ...


Keywords: compiled simulation, instruction abstraction, instruction set architectures, interpretive simulation

4 SIMP (Single Instruction stream/Multiple instruction Pipelining): a novel high-speed single-processor architecture

 K. Murakami, N. Irie, S. Tomita


April 1989 **ACM SIGARCH Computer Architecture News , Proceedings of the 16th annual international symposium on Computer architecture ISCA '89**, Volume 17 Issue 3

Publisher: ACM Press

Full text available:  [pdf\(1.23 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)


SIMP is a novel multiple instruction-pipeline parallel architecture. It is targeted for enhancing the performance of SISD processors drastically by exploiting both temporal and spatial parallelisms, and for keeping program compatibility as well. Degree of performance enhancement achieved by SIMP depends on; i) how to supply multiple instructions continuously, and ii) how to resolve data and control dependencies effectively. We have devised the outstanding techniques for instruction fetch an ...

5 Instruction generation for hybrid reconfigurable systems

 R. Kastner, A. Kaplan, S. Ogrenci Memik, E. Bozorgzadeh

October 2002 **ACM Transactions on Design Automation of Electronic Systems (TODAES)**, Volume 7 Issue 4


Publisher: ACM Press

Full text available:  [pdf\(538.25 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Future computing systems need to balance flexibility, specialization, and performance in order to meet market demands and the computing power required by new applications. Instruction generation is a vital component for determining these trade-offs. In this work, we present theory and an algorithm for instruction generation. The algorithm profiles a dataflow graph and iteratively contracts edges to create the templates. We discuss how to target the algorithm toward the novel problem of instructi ...

Keywords: FPGA, high-level synthesis, reconfigurable computing

6 A time-stamping algorithm for efficient performance estimation of superscalar processors

 Gabriel Loh

June 2001 **ACM SIGMETRICS Performance Evaluation Review , Proceedings of the 2001 ACM SIGMETRICS international conference on Measurement and modeling of computer systems SIGMETRICS '01**, Volume 29 Issue 1

Publisher: ACM Press

Full text available:  [pdf\(1.11 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)

The increasing complexity of modern superscalar microprocessors makes the evaluation of new designs and techniques much more difficult. Fast and accurate methods for simulating program execution on realistic and hypothetical processor models are of great interest to many computer architects and compiler writers. There are many existing techniques, from profile based runtime estimation to complete cycle-level simulations. Many researchers choose to sacrifice the speed of profiling for the accurac ...

7 A Complexity-Effective Approach to ALU Bandwidth Enhancement for Instruction-Level Temporal Redundancy



Angshuman Parashar, Sudhanva Gurumurthi, Anand Sivasubramaniam

March 2004 **ACM SIGARCH Computer Architecture News , Proceedings of the 31st annual international symposium on Computer architecture ISCA '04**,
Volume 32 Issue 2

Publisher: IEEE Computer Society, ACM Press

Full text available: [pdf\(450.01 KB\)](#) Additional Information: [full citation](#), [abstract](#)

Previous proposals for implementing instruction-level temporal redundancy in out-of-order cores have reported a performance degradation of upto 45% in certain applications compared to an execution which does not have any temporal redundancy. An important contributor to this problem is the insufficient number of ALUs for handling the amplified load injected into the core. At the same time, increasing the number of ALUs can increase the complexity of the issue logic, which has been pointed out to be oneo ...

Keywords: Complexity-effective design, Instruction Reuse, Temporal Redundancy

8 Micro-architectural techniques: Instruction packing: reducing power and delay of the dynamic scheduling logic



Joseph J. Sharkey, Dmitry V. Ponomarev, Kanad Ghose, Oguz Ergin

August 2005 **Proceedings of the 2005 international symposium on Low power electronics and design ISLPED '05**

Publisher: ACM Press

Full text available: [pdf\(239.57 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

The instruction scheduling logic used in modern superscalar microprocessors often relies on associative searching of the issue queue entries to dynamically wakeup instructions for the execution. Traditional designs use one issue queue entry for each instruction, regardless of the actual number of operands actively used in the wakeup process. In this paper we propose *Instruction Packing* - a novel microarchitectural technique that reduces both the delay and the power consumption of the issue ...

Keywords: instruction packing, issue queue, low power

9 Instruction merging and specialization in the SICStus Prolog virtual machine



Henrik Nässén, Mats Carlsson, Konstantinos Sagonas

September 2001 **Proceedings of the 3rd ACM SIGPLAN international conference on Principles and practice of declarative programming**

Publisher: ACM Press

Full text available: [pdf\(249.88 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Wanting to improve execution speed and reduce code size of SICStus Prolog programs, we embarked on a project whose aim was to systematically investigate combination and specialization of WAM instructions. Various variants of the SICStus Prolog virtual machine instruction set were designed, implemented, and their performance was evaluated against standard benchmarks and on big Prolog programs. In this paper, we describe our

methodology in finding appropriate candidates for instruction merging and ...

10 Rapid Configuration and Instruction Selection for an ASIP: A Case Study

Newton Cheung, Jorg Henkel, Sri Parameswaran

March 2003 **Proceedings of the conference on Design, Automation and Test in Europe - Volume 1 DATE '03**

Publisher: IEEE Computer Society

Full text available:  [pdf\(447.59 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [index terms](#)

 [Publisher Site](#)

We present a methodology that maximizes the performance of Tensilica based Application Specific Instruction-set Processor (ASIP) through instruction selection when an area constraint is given. Our approach rapidly selects from a set of pre-fabricated coprocessors/functional units from our library of pre-designed specific instructions (to evaluate our technology we use the Tensilica platform). As a result, we significantly increase application performance while area constraints are satisfied. Our ...

11 Worst-case execution time analysis on modern processors

 Kelvin D. Nilsen, Bernt Rygg


November 1995 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 1995 workshop on Languages, compilers, & tools for real-time systems LCTES '95**, Volume 30 Issue 11

Publisher: ACM Press

Full text available:  [pdf\(998.42 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)


Many of the trends that have dominated recent evolution and advancement within the computer architecture community have complicated the analysis of task execution times. Most of the difficulties result from two particular emphases: (1) Instruction-level parallelism, and (2) Optimization of average-case behavior rather than worst-case latencies. Both of these trends have resulted in increased nondeterminism in the time required to execute particular code sequences. And since the analysis required ...

12 ILP-based Instruction Scheduling for IA-64

 Daniel Kästner, Sebastian Winkel


August 2001 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN workshop on Languages, compilers and tools for embedded systems LCTES '01 , Proceedings of the 2001 ACM SIGPLAN workshop on Optimization of middleware and distributed systems OM '01**, Volume 36 Issue 8

Publisher: ACM Press

Full text available:  [pdf\(369.18 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

The IA-64 architecture has been designed as a synthesis of VLIW and superscalar design principles. It incorporates typical functionality known from embedded processors as multiply/accumulate units and SIMD operations for 3D graphics operations. In this paper we present an ILP formulation for the problem of instruction scheduling for IA-64. In order to obtain a feasible schedule it is necessary to model the data dependencies, resource constraints as well as additional encoding restrictions&mdas ...

13 Instruction-level test methodology for CPU core self-testing

 Saeed Shamshiri, Hadi Esmaeilzadeh, Zainalabdein Navabi

October 2005 **ACM Transactions on Design Automation of Electronic Systems (TODAES)**, Volume 10 Issue 4

Publisher: ACM Press

Full text available:  [pdf\(2.26 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

TIS is an instruction-level methodology for processor core self-testing that enhances

instruction set of a CPU with test instructions. Since the functionality of test instructions is the same as the NOP instruction, NOP instructions can be replaced with test instructions. Online testing can be accomplished without any performance penalty. TIS tests different parts of the processor and detects stuck-at faults. This method can be employed in offline and online testing of single-cycle, multicycle a ...

Keywords: BIST, CPU core testing, Instruction level testing, pipelined processor, software-based self testing, test instruction set


14 Instruction-level DFT for testing processor and IP cores in system-on-a-chip



Wei-Cheng Lai, Kwang-Ting Cheng

June 2001 **Proceedings of the 38th conference on Design automation**

Publisher: ACM Press

Full text available:  [pdf\(75.03 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Self-testing manufacturing defects in a system-on-a-chip (SOC) by running test programs using a programmable core has several potential benefits including, at-speed test-ing, low DFT overhead due to elimination of dedicated test circuitry and better power and thermal management during testing. However, such a self-test strategy might require a lengthy test program and might achieve a high enough fault coverage. We propose a DFT methodology to improve the fault coverage and reduce the test p ...

15 LLVA: A Low-level Virtual Instruction Set Architecture



Vikram Adve, Chris Lattner, Michael Brukman, Anand Shukla, Brian Gaeke

December 2003 **Proceedings of the 36th annual IEEE/ACM International Symposium on Microarchitecture**

Publisher: IEEE Computer Society

Full text available:  [pdf\(196.08 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [index terms](#)

A virtual instruction set architecture (V-ISA) implemented via a processor-specific software translation layer can provide great flexibility to processor designers. Recent examples such as Crusoe and DAISY, however, have used existing hardware instruction sets as virtual ISAs, which complicates translation and optimization. In fact, there has been little research on specific designs for a virtual ISA for processors. This paper proposes a novel virtual ISA (LLVA) and a translation strategy for implementi ...

16 A control program for Computer Assisted Instruction on a general purpose computer



Eugene G. Kerr, T. C. Ting, William E. Walden

August 1969 **Proceedings of the 1969 24th national conference**

Publisher: ACM Press

Full text available:  [pdf\(401.55 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)




A Computer Assisted Instruction system, running in a multiprogramming environment on an IBM 360 Model 67, is described. Instructional sequences can be built from basic units called frames. These frames can be presented for keypunching in a format which requires no computer programming knowledge. The frames are then processed and stored in direct access files where they can be easily expanded or modified. The system is being used to provide mathematics instruction to students in small remote ...

17 GT-EP: a novel high-performance real-time architecture



Wei Siong Tan, H. Russ, Cecil O. Alford

April 1991 **ACM SIGARCH Computer Architecture News , Proceedings of the 18th annual international symposium on Computer architecture ISCA '91**, Volume 19 Issue 3

Publisher: ACM PressFull text available:  [pdf\(1.09 MB\)](#)Additional Information: [full citation](#), [references](#), [index terms](#)**18** Space-time scheduling of instruction-level parallelism on a raw machine Walter Lee, Rajeev Barua, Matthew Frank, Devabhaktuni Srikrishna, Jonathan Babb, Vivek Sarkar, Saman AmarasingheOctober 1998 **ACM SIGPLAN Notices , ACM SIGOPS Operating Systems Review , Proceedings of the eighth international conference on Architectural support for programming languages and operating systems ASPLOS-VIII**, Volume 33 , 32 Issue 11 , 5**Publisher:** ACM PressFull text available:  [pdf\(1.79 MB\)](#)Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Increasing demand for both greater parallelism and faster clocks dictate that future generation architectures will need to decentralize their resources and eliminate primitives that require single cycle global communication. A Raw microprocessor distributes all of its resources, including instruction streams, register files, memory ports, and ALUs, over a pipelined two-dimensional mesh interconnect, and exposes them fully to the compiler. Because communication in Raw machines is distributed, com ...

19 Low power design for embedded and real-time systems: Instruction scheduling of VLIW architectures for balanced power consumption Shu Xiao, Edmund M-K. LaiJanuary 2005 **Proceedings of the 2005 conference on Asia South Pacific design automation ASP-DAC '05****Publisher:** ACM PressFull text available:  [pdf\(408.47 KB\)](#)Additional Information: [full citation](#), [abstract](#), [references](#)

An instruction word in VLIW (very long instruction word) processors consists of a variable number of individual instructions. Therefore the power consumption variation over time significantly depends on the parallel instruction schedule generated by the compiler. Sharp power variations across time cause power supply noises, degrade chip reliability and accelerate battery exhaustion. This paper proposes a branch and bound algorithm for instruction scheduling of VLIW architectures that effectively ...

20 Instruction fetch mechanisms for multipath execution processors

Artur Klauser, Dirk Grunwald

November 1999 **Proceedings of the 32nd annual ACM/IEEE international symposium on Microarchitecture****Publisher:** IEEE Computer SocietyFull text available:  [pdf\(1.43 MB\)](#)Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)
[Publisher Site](#)

Branch mispredictions can have a major performance impact on high-performance processors. Multipath execution has recently been introduced to help limit the misprediction penalties incurred by branches that are difficult to predict. This paper presents efficient instruction fetch architecture designs for these multipath processor execution cores. We evaluate a number of design trade-offs for the first-level instruction cache and the multipath PC fetch arbiter. Furthermore we evaluate the e ...

Useful downloads:  [Adobe Acrobat](#)  [QuickTime](#)  [Windows Media Player](#)  [Real Player](#)